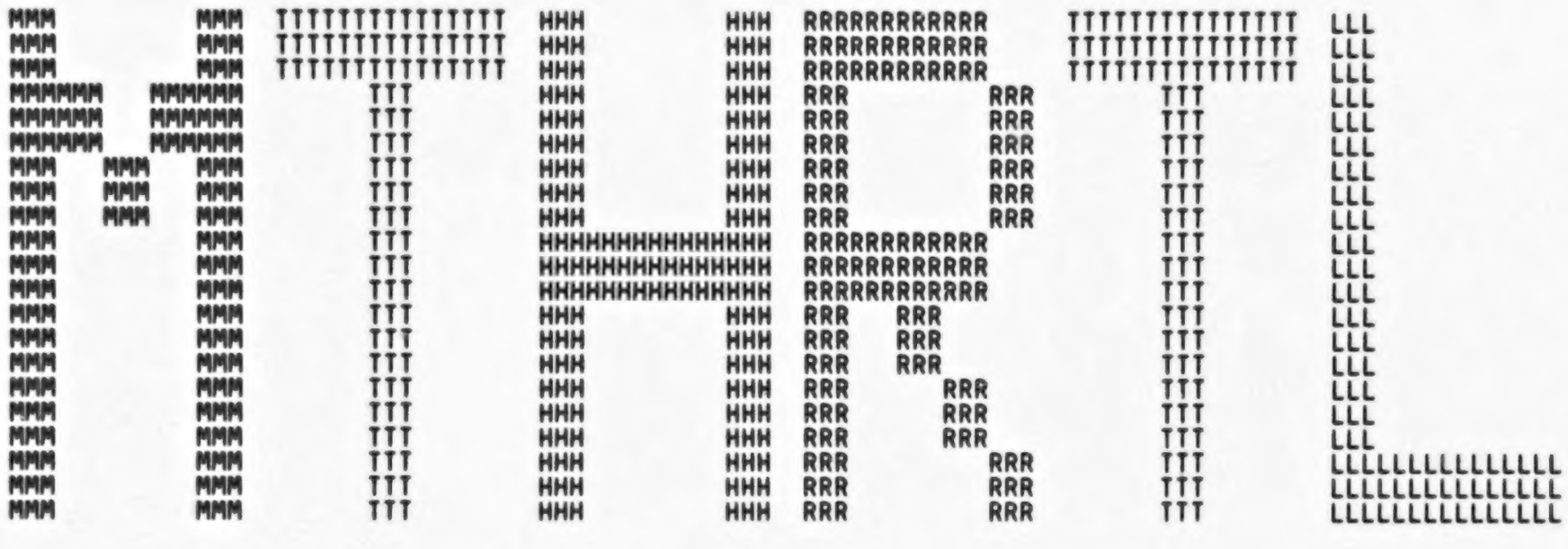
MMM MMM TTTTTTTTTTTTTTT HHH HHH RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
MMMMMM MMMMMM TTT HHH HHH RRR RRR TTT LLL
MMM MMM MMM TTT HHH HHH RRR RRR TTT LLL
MMM MMM MMM TTT HHHHHHHHHHHHHHH RRRRRRRRRRR TTT LLL
MMM MMM MMM TTT HHH HHH RRR RRR TTT LLL
MMM MMM MMM TTT HHH HHH RRR RRR TTT LLL
MMM MMM MMM TTT HHH HHH RRR RRR TTT LLLLLLLLLLLLLLL

FILEID**MTHSINH

```
MM      MM  TTTTTTTTTT  HH      HH    SSSSSSSS    IIIIII    NN      NN  HH      HH
MM      MM  TTTTTTTTTT  HH      HH    SSSSSSSS    IIIIII    NN      NN  HH      HH
MMMM  MMMM      TT      HH      HH  SS              II      NN      NN  HH      HH
MMMM  MMMM      TT      HH      HH  SS              II      NN      NN  HH      HH
MM  MM  MM      TT      HH      HH  SS              II      NNNN    NN  HH      HH
MM  MM  MM      TT      HH      HH  SS              II      NNNN    NN  HH      HH
MM      MM      TT      HHHHHHHHHH    SSSSSS        II      NN  NN  NN  HHHHHHHHHH
MM      MM      TT      HHHHHHHHHH    SSSSSS        II      NN  NN  NN  HHHHHHHHHH
MM      MM      TT      HH      HH          SS      II      NN    NNNN  HH      HH
MM      MM      TT      HH      HH          SS      II      NN    NNNN  HH      HH
MM      MM      TT      HH      HH          SS      II      NN      NN  HH      HH      ....
MM      MM      TT      HH      HH          SS      II      NN      NN  HH      HH      ....
MM      MM      TT      HH      HH  SSSSSSSS      IIIIII    NN      NN  HH      HH      ....
MM      MM      TT      HH      HH  SSSSSSSS      IIIIII    NN      NN  HH      HH      ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

```
0000     1          .TITLE  MTH$SINH        ; Floating Point Hyperbolic Sine routine
0000     2                                  ; (SINH)
0000     3          .IDENT /1-007/          ; File: MTHSINH.MAR  Edit: RNH1007
0000     4 ;
0000     5 ;*****************************************************************************
0000     6 ;*                                                                           *
0000     7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9 ;*  ALL RIGHTS RESERVED.                                                     *
0000    10 ;*                                                                           *
0000    11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    16 ;*  TRANSFERRED.                                                             *
0000    17 ;*                                                                           *
0000    18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    20 ;*  CORPORATION.                                                             *
0000    21 ;*                                                                           *
0000    22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24 ;*                                                                           *
0000    25 ;*                                                                           *
0000    26 ;*****************************************************************************
0000    27 ;
0000    28 ;
0000    29 ; FACILITY: MATH LIBRARY
0000    30 ;++
0000    31 ; ABSTRACT:
0000    32 ;
0000    33 ; MTH$SINH is a function  which  returns the floating point hyperbolic sine
0000    34 ; of its single precision floating point argument. The call is standard
0000    35 ; call-by-reference.
0000    36 ;
0000    37 ;--
0000    38 ;
0000    39 ; VERSION: 01
0000    40 ;
0000    41 ; HISTORY:
0000    42 ; AUTHOR:
0000    43 ;       Peter Yuo, 29-Jun-77: Version 01
0000    44 ;
0000    45 ; MODIFIED BY:
0000    46 ;
0000    47 ;
0000    48 ;
```

```
0000     50          .SBTTL  HISTORY ; Detailed Current Edit History
0000     51
0000     52
0000     53 ; ALGORITHMIC DIFFERENCES FROM FP-11/C ROUTINE: none
0000     54 ;
0000     55 ; Edit History for Version 01 of MTH$SINH
0000     56 ;
0000     57 ; 0-2   MTH$$ERROR changed to MTH$$SIGNAL.
0000     58 ;       MTH$_... changed to MTH__....
0000     59 ;       Changed error handling mechanism. Put error result in R0 before
0000     60 ;       calling MTH$$SIGNAL in order to allow user modify error result.
0000     61 ;
0000     62 ; 0-3   Six term Taylor series, in powers of the argument, replaced
0000     63 ;       by four term Chebyshev series, in powers of ARG**2, with
0000     64 ;       overhang, for small arguments; this improves accuracy.
0000     65 ;       18-May-1978; Mary Payne
0000     66 ; 1-001 - Update version number and copyright notice.  JBS 16-NOV-78
0000     67 ; 1-002 - Change MTH__FLOOVEMAT to MTH$K_FLOOVEMAT.  JBS 07-DEC-78
0000     68 ; 1-003 - Remove $SRMDEF macro - not needed.  JBS 16-DEC-78
0000     69 ; 1-004 - Add "_" to the PSECT directive.  JBS 22-DEC-78
0000     70 ; 1-005 - Declare externals.  SBL 17-May-1979
0000     71 ; 1-006 - Changed lower limit for Chebyshev approximation from 2**-11 to
0000     72 ;         2**-12.
0000     73 ;       - Eliminated second call to EXP for input values between 12.5*ln2
0000     74 ;         and 127*ln2.
0000     75 ;       - Changed all final floating point divisions by 2 to interger
0000     76 ;         subtracts of 1 from the exponent field.
0000     77 ;       - Extended maximum range from 87.69 to 128*ln2=88.72.
0000     78 ;       - Chaneged logic for computing EXP(|x|-ln2) to reduce error.
0000     79 ;       - RNH 10-FEB-81
0000     80 ; 1-007 - Change W^ to G^ on call to MTH$$SIGNAL RNH 09-Sept-1981
```

```
                                        .SBTTL  DECLARATIONS    ; Declarative Part of Module

                                ;
                                ; INCLUDE FILES:        MTHJACKET.MAR
                                ;

                                ;
                                ; EXTERNAL SYMBOLS:
                                ;
                                        .DSABL  GBL
                                        .EXTRN  MTH$EXP_R4
                                        .EXTRN  MTH$K_FLOOVEMAT
                                        .EXTRN  MTH$$SIGNAL

                                ;
                                ; EQUATED SYMBOLS:

0F3343B0                        LF_127_LOG_2    = ^X0F3343B0    ; 127*ln2
721643B1                        LF_128_LOG_2    = ^X721643B1    ; 128*ln2-2**-24
A123420A                        LF_12.5_LOG_2   = ^XA123420A    ; 12.5*ln2
72804031                        LF_LOG_2_HI     = ^X72804031    ; (high 17 bits of ln2) + 2**-17
1100B7D0                        LF_LOG_2_LO     = ^X1100B7D0    ; ln2 - LF_LOG_2_HI
00004080                        SF_1.0  = ^F1.0                 ; 1.0
00000004                        value   = 4                     ; value.rf.r

                                ;
                                ; MACROS:       none
                                ;
                                ;
                                ; PSECT DECLARATIONS:

                                        .PSECT  _MTH$CODE       PIC,SHR,LONG,EXE,NOWRT
                                                                ; program section for math routines
                                ;
                                ; OWN STORAGE:  none
                                ;
                                ;
                                ; CONSTANTS:
                                ;

                                SINHTAB:
698B 3A50                               .WORD   ^0035120,^0064613       ; DECIMAL:     0.1987574E-03
887A 3D08                               .WORD   ^0036410,^0104172       ; DECIMAL:     0.8333320E-02
AAAB 3F2A                               .WORD   ^0037452,^0125253       ; DECIMAL:     0.1666667E+00
0000 0000                               .WORD   0,0                     ; DECIMAL: 0.D0
 00000004               SINHLEN = .- SINHTAB/4
```

```
                        0010   131          .SBTTL  MTH$SINH  - Standard Single Precision Floating SINH
                        0010   132
                        0010   133
                        0010   134 ;++
                        0010   135 ; FUNCTIONAL DESCRIPTION:
                        0010   136 ;
                        0010   137 ; SINH  - single precision floating point function
                        0010   138 ;
                        0010   139 ; SINH(X) is computed as:
                        0010   140 ;
                        0010   141 ;       If |X| < 2**-12, SINH(X) = X.
                        0010   142 ;       IF 2**-12 =< |X| < 0.25, SINH(X) = Chebyshev Series
                        0010   143 ;       If 0.25 =< |X| < 12.5*ln2, SINH(X) = (EXP(X) - EXP(-X))/2
                        0010   144 ;       If 12.5=< |X| < 127*ln2, SINH(X) = sign(X)*EXP(|X|)/2
                        0010   145 ;       If 127*ln2 =< |X| < 128*ln2, then SINH(X) = sign(X)* EXP(|X|-LOG(2)).
                        0010   146 ;       If 128*ln2 =< |X|, then overflow.
                        0010   147 ;
                        0010   148 ; CALLING SEQUENCE:
                        0010   149 ;
                        0010   150 ;       SINH.wf.v = MTH$SINH(x.rf.r)
                        0010   151 ;
                        0010   152 ; INPUT PARAMETERS:
                        0010   153
                00000004 0010   154          LONG = 4                           ; define longword multiplier
                00000004 0010   155          x = 1 * LONG                       ; Contents of x is the argument
                        0010   156
                        0010   157 ; IMPLICIT INPUTS:      none
                        0010   158 ;
                        0010   159 ; OUTPUT PARAMETERS:
                        0010   160 ;
                        0010   161 ;       VALUE:  floating hyperbolic sine of the argument
                        0010   162 ;
                        0010   163 ; IMPLICIT OUTPUTS:     none
                        0010   164 ;
                        0010   165 ; COMPLETION CODES:     none
                        0010   166 ;
                        0010   167 ; SIDE EFFECTS:
                        0010   168 ;
                        0010   169 ; Signal: MTH$_FLOOVEMAT if 128*ln2 =< |X| with reserved operand in R0 (copied
                        0010   170 ; to the signal mechanism vector CHF$L_MCH_R0/R1 by LIB$SIGNAL). Associated
                        0010   171 ; message is: "FLOATING OVERFLOW IN MATH LIBRARY". Result is reserved operand -0.0
                        0010   172 ; unless a user supplied (or any) error handler changes CHF$L_MCH_R0/R1.
                        0010   173 ;
                        0010   174 ; NOTE: This procedure disables floating point underflow, enables integer
                        0010   175 ; overflow.
                        0010   176 ;
                        0010   177 ;---
                        0010   178
                        0010   179
                   403C 0010   180          .ENTRY  MTH$SINH, ^M<IV, R2, R3, R4, R5>
                        0012   181                                             ; standard call-by-reference entry
                        0012   182                                             ; disable DV (and FU), enable IV
                        0012   183          MTH$FLAG_JACKET                    ; flag that this is a jacket procedure in
                        0012
6D  00000000'GF   9E    0012                 MOVAB   G^MTH$$JACKET_HND, (FP)
                        0019                                                    ; set handler address to jacket
                        0019                                                    ; handler
```

```
                                0019
                                0019    184                             ; case of an error in routine
                                0019    185                             ; If an error, convert signal to user PC
                                0019    186                             ; and resignal
      55    04 BC    50        0019    187             MOVF    @value(AP), R5          ; R5 = X = @value(AP)
      50    55    50            001D    188             MOVF    R5, R0
   50    8000 8F    AA          0020    189             BICW2   #^X8000, R0             ; R0 = |X|
 3F80 8F         50    B1       0025    190             CMPW    R0, #^X3F80             ; compare |X| with 0.25
                 1A    18       002A    191             BGEQ    GEQ_TO_0.25             ; branch if |X| >= 0.25
                                002C    192
                                002C    193 ;
                                002C    194 ; |X| < 0.25
                                002C    195 ;
                                002C    196
 3A80 8F         50    B1       002C    197             CMPW    R0, #^X3A80             ; compare |X| with 2**-12
                 04    18       0031    198             BGEQ    GEQ_TO_2M12             ; branch if |X| >= 2**-12
                                0033    199
                                0033    200 ;
                                0033    201 ; |X| < 2**-12
                                0033    202 ;
                                0033    203
      50    55    50            0033    204             MOVF    R5, R0                  ; R0 = X
                 04            0036    205             RET                             ; return with result = Argument
                                0037    206
                                0037    207 ;
                                0037    208 ; 2**-12 =< |X| < 0.25
                                0037    209 ;
                                0037    210
                                0037    211 GEQ_TO_2M12:
      50    50    44            0037    212             MULF    R0, R0                  ; Get ARG**2 for POLYF
 C1 AF    03    50    55        003A    213             POLYF   R0, #SINHLEN-1, SINHTAB ; R0 = SUM(Ci*X**(2*i))
                                003F    214                                             ; Last coefficient is zero
      50    55    44            003F    215             MULF    R5, R0                  ; MUL by ARG and then
      50    55    40            0042    216             ADDF    R5, R0                  ; add in ARG with overhang
                 04            0045    217             RET                             ; return with result in R0
                                0046    218
                                0046    219 ;
                                0046    220 ; 0.25 =< |X|
                                0046    221 ;
                                0046    222
                                0046    223 GEQ_TO_0.25:
0F3343B0 8F         50    51    0046    224             CMPF    R0, #LF_127_LOG_2               ; compare |X| with 127*ln2
                 3A    14       004D    225             BGTR    GTR_THAN_127_LOG_2              ; branch if X > 127*ln2
                                004F    226
                                004F    227 ;
                                004F    228 ; 0.25 =< |X| =< 127*ln2
                                004F    229 ;
                                004F    230
A123420A 8F         50    51    004F    231             CMPF    R0, #LF_12.5_LOG_2      ; Compare |X| to 12.5*ln2.  If greater,
                 1E    14       0056    232             BGTR    ONE_TERM_ONLY           ; only one call to EXP is needed
                                0058    233 ;
                                0058    234 ; 0.25 =< |X| < 12.5*ln2
                                0058    235 ;
                                0058    236
      50    55    50            0058    237             MOVF    R5, R0                  ; R0 = X
00000000'EF         16          005B    238             JSB     MTH$EXP_R4              ; R0 = EXP(X)
                    50    DD    0061    239             PUSHL   R0                      ; push EXP(X) on stack
```

```
                50    55    52  0063   240          MNEGF   R5, R0                  ; R0 = -X
      00000000'EF    16  0066   241          JSB     MTH$EXP_R4               ; R0 = EXP(-X)
          50    8E    50    43  006C   242          SUBF3   R0, (SP)+, R0            ; R0 = EXP(X) - EXP(-X)
          50  0080  8F    A2  0070   243          SUBW    #^X0080, R0              ; R0 = (EXP(X) - EXP(-X))/2
                      04  0075   244          RET                              ; return with result in R0
                          0076   245
                          0076   246 ;
                          0076   247 ; 12.5 =< |x| < 127*ln2
                          0076   248 ;
                          0076   249
                          0076   250 ONE_TERM_ONLY:
      00000000'EF    16  0076   251          JSB     MTH$EXP_R4               ; R0 = EXP(|X|)
                55    53  007C   252          TSTF    R5                       ; If X is negative, change
                      03  007E   253          BGTR    POSITIVE                 ; the sign of EXP(|X|)
                50    50  0080   254          MNEGF   R0,R0                    ;
                          0083   255 POSITIVE:                                ; R0 = sign(X)*EXP(|X|)
          50  0080  8F    A2  0083   256          SUBW    #^X0080, R0              ; R0 = sign(X)*EXP(|X|)/2
                      04  0088   257          RET
                          0089   258
                          0089   259 ;
                          0089   260 ; 127*ln2 =< |X|
                          0089   261 ;
                          0089   262
                          0089   263 GTR_THAN_127_LOG_2:
    721643B1  8F    50    51  0089   264          CMPF    R0, #LF_128_LOG_2        ; Compare |X| with 128*ln2. If greater,
                      20    18  0090   265          BGEQ    ERROR                    ; overflow is signaled
                          0092   266
                          0092   267 ;
                          0092   268 ; 127*ln2 =< |X| < 128*ln2
                          0092   269 ;
                          0092   270
          50    72804031  8F    42  0092   271          SUBF    #LF_LOG_2_HI, R0         ; R0 = |X|-(high order bits of ln2)
                00000000'EF    16  0099   272          JSB     MTH$EXP_R4               ; R0 = EXP(|X|-(high order bits of ln2)
    51    50    1100B7D0  8F    45  009F   273          MULF3   #LF_LOG_2_LO, R0, R1     ;
                    50    51    42  00A7   274          SUBF    R1, R0                   ; R0 = EXP (|X| - ln2)
                          55    53  00AA   275          TSTF    R5                       ; test the sign of X
                          03    18  00AC   276          BGEQ    10$                      ; branch if X >= 0
                    50    50    52  00AE   277          MNEGF   R0, R0                   ; R0 = sign(X) * EXP(|X|-LOG(2))
                          04    04  00B1   278 10$:      RET                              ; return with result in R0
                                    00B2   279
                                    00B2   280 ;
                                    00B2   281 ; 128*ln2 =< |X|, error
                                    00B2   282 ;
                                    00B2   283
              7E    00'8F    9A  00B2   284 ERROR:  MOVZBL  #MTH$K_FLOOVEMAT, -(SP) ; condition value
            50    01    0F    78  00B6   285          ASHL    #15, #T, R0              ; R0 = result = reserved operand -0.0
                                    00BA   286                                   ; goes to signal mechanism vector
                                    00BA   287                                   ; (CHF$L_MCH_R0/R1) so error handler
                                    00BA   288                                   ; can modify the result.
      00000000'GF    01    FB  00BA   289          CALLS   #1, G^MTH$$SIGNAL        ; signal error and use real user's PC
                                    00C1   290                                   ; independent of CALL vs JSB
                          04    04  00C1   291          RET                              ; return - R0 restored from CHF$L_MCH_R0/R1
                                    00C2   292
                                    00C2   293
                                    00C2   294
                                    00C2   295          .END
```

```
ERROR                           000000B2 R      01
GEQ_TO_0.25                     00000046 R      01
GEQ_TO_2M12                     00000037 R      01
GTR_THAN_127_LOG_2              00000089 R      01
LF_T2.5_LOG_2                 = A123420A
LF_127_LOG_2                  = 0F3343B0
LF_128_LOG_2                  = 721643B1
LF_LOG_2_HI                   = 72804031
LF_LOG_2_LO                   = 1100B7D0
LONG                          = 00000004
MTH$$JACKET_HND                 ********    X   01
MTH$$SIGNAL                     ********    X   00
MTH$EXP_R4                      ********    X   00
MTH$K_FLOOVEMAT                 ********    X   00
MTH$SINH                        00000010 RG     01
ONE_TERM_ONLY                   00000076 R      01
POSITIVE                        00000083 R      01
SINHLEN                       = 00000004
SINHTAB                         00000000 R      01
VALUE                         = 00000004
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
_MTH$CODE	000000C2 (194.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.07	00:00:01.24
Command processing	119	00:00:00.62	00:00:03.55
Pass 1	85	00:00:00.89	00:00:05.18
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	62	00:00:00.74	00:00:03.14
Symbol table output	4	00:00:00.02	00:00:00.78
Psect synopsis output	3	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	306	00:00:02.38	00:00:13.93

The working set limit was 900 pages.
4456 bytes (9 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 22 non-local and 1 local symbols.
355 source lines were read in Pass 1, producing 11 object records in Pass 2.
1 page of virtual memory was used to define 1 macro.

```
                        +---------------------------+
                        ! Macro library statistics !
                        +---------------------------+

Macro library name                      Macros defined
------------------                      --------------
_$255$DUA28:[SYSLIB]STARLET.MLB;2                0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS$:MTHSINH/OBJ=OBJ$:MTHSINH MSRC$:MTHJACKET/UPDATE=(ENH$:MTHJACKET)+MSRC$:
```

MTHJMIN0
LIS

MTHMSG
LIS

MTHIMIN0
LIS

MTHMAX1
LIS

MTHJIDNNT
LIS

MTHMSGDEF
LIS

MTHJIHNNT
LIS

MTHSGN
LIS

MTHJMAX0
LIS

MTHMOD
LIS

MTHSIGNAL
LIS

MTHSQRTR2
LIS

MTHJNINT
LIS

MTHSINCOS
LIS

MTHSQRT
LIS

MTHININT
LIS

MTHRANDOM
LIS

MTHJIGNNT
LIS

MTHMINI
LIS

MTHJISIGN
LIS

MTHSIGN
LIS

MTHSINH
LIS